Biuletyn WAT
Vol. LX, Nr 4, 2011

Projekt funkcji skrótu bazującej na konstrukcji MCM i przekształceniach trójkątnych

MICHAŁ GLET

Wojskowa Akademia Techniczna, Wydział Cybernetyki,
Instytut Matematyki i Kryptologii, 00-908 Warszawa, ul. S. Kaliskiego 2

Streszczenie. Praca przedstawia autorskie podejście do problemu projektowania funkcji skrótu. Zaproponowana konstrukcja bazuje na schemacie Mix-Compress-Mix, przekształceniach trójkątnych oraz multipermutacjach. Dzięki temu przedstawiona rodzina funkcji skrótu charakteryzuje się dobrymi własnościami kryptograficznymi oraz możliwością efektywnej implementacji w środowiskach wielowątkowych.
Słowa kluczowe: funkcja skrótu, multipermutacje, Mix-Compress-Mix, przekształcenia trójkątne, MD

1. Wstęp

Dzisiejszy świat pełny jest elektroniki, a otaczająca nas rzeczywistość w dużej mierze kształtowana jest przez zdobycze techniki z ostatnich kilku dekad. Nikt nie wyobraża sobie życia bez telewizora czy telefonu. Praktycznie każdy z nas ma w domu komputer, telewizję cyfrową, dostęp do Internetu i posiada telefon komórkowy. Coraz większa część społeczeństwa za pośrednictwem komputera podłączonego do Internetu załatwia sprawy, które jeszcze kilka lat temu trzeba było załatwiać osobiście, niejednokrotnie tracąc czas, stojąc w kolejkach. Przykładami tej sytuacji mogą być banki internetowe, elektroniczne markety, aukcje. Dzisiaj nie musimy już nigdzie chodzić, gdy chcemy coś kupić lub zrobić przelew. Wystarczy, że włączymy komputer, połączymy się z Internetem i skorzystamy z dostępnych usług online. No właśnie, dziś każdy może skorzystać z tego typu usług na prak-

tycznie dowolnym komputerze, w dowolnym miejscu na świecie. A skąd ten ktoś wie, że, a może czy, jest bezpieczny? Czy dane przesyłane do banków nie zostaną podejrzane przez sąsiada? Czy sąsiad nie zmieni na przykład kwoty przelewu lub numeru rachunku?

Świat mediów elektronicznych musi, z natury świadczonych usług, mieć możliwość zapewnienia bezpieczeństwa wszystkim uczestnikom transakcji. Istnieje wiele różnych sposobów podejścia do kwestii bezpieczeństwa, wiele różnych metodologii bezpieczeństwa, wiele różnych schematów i protokołów, które przy odwrotnych zastosowaniach mają zapewniać bezpieczeństwo. Praktycznie każdy taki mechanizm, w większej lub mniejszej mierze, korzysta z prymitywów kryptograficznych. Wykorzystywane jest wszystko, co dała kryptografia – od szyfrów symetrycznych i asymetrycznych, poprzez generatory pseudolosowe po funkcje skrótu, kody korekcyjne i funkcje sum kontrolnych. Nie sposób nie wspomnieć o bardziej złożonych mechanizmach kryptograficznych opartych o te prymitywy, jak choćby funkcje typu MAC, HMAC czy AE. Kryptografia daje bazowe mechanizmy, które można na przykład wykorzystać do zapewnienia poufności, autentyczności, autoryzacji czy uwierzytelnienia.

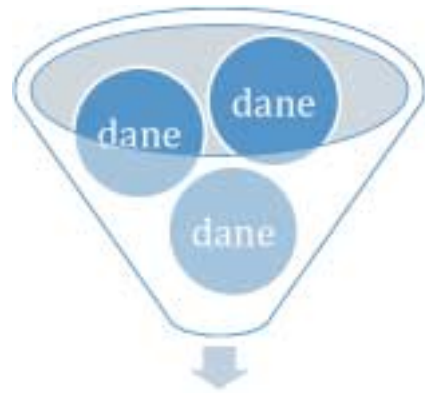
Myślę, że mało jest na świecie ludzi, którzy nie słyszeliby słowa szyfr, jak również mało jest na świecie ludzi, którzy słyszeliby nazwę funkcja skrótu, a jeszcze mniej tych, którzy wiedzą, czym ona jest i jak ją prawidłowo stosować. Sytuacja taka wynika z faktu, iż o szyfrach słyszymy praktycznie od młodości. Powstało wiele ogólnie dostępnych publikacji, książek beletrystycznych, komiksów i filmów, w których przewija się szyfr jako coś zapewniającego tajemniczość i grozę. W realnym świecie jest tak, że tam gdzie wykorzystuje się szyfr, czy to symetryczny, czy asymetryczny, zazwyczaj wykorzystuje się również funkcję skrótu. Prymitywy te odgrywają niezwykle istotną rolę w zapewnianiu bezpieczeństwa, szczególnie uwzględniając aspekty autoryzacji, uwierzytelnienia i niezaprzeczalności danych i osób.

2. Czym są funkcje skrótu

Czym więc jest funkcja skrótu? Funkcję skrótu porównać można do odcisku palca, czy, co w dzisiejszym świecie jest bardziej popularne, do śladu DNA. Teoretycznie każdy człowiek ma inny kod DNA, po którym można go jednoznacznie określić i rozpoznać. Posiadając czyjąś próbkę DNA, można wykazać, że należy ona do tej właśnie osoby i jest to uznawane w świetle prawa za przyporządkowanie jednoznaczne. Podobnie sprawy mają się w odniesieniu do funkcji skrótu. Jest to funkcja, która działając na danych, wytwarza coś podobnego do próbki DNA – skrót. Identyfikuje on

jednoznacznie dane, dla których został utworzony. Posiadając taki skrót i dane, z bardzo dużym prawdopodobieństwem możemy stwierdzić, gdy skrót pasuje do danych, że właśnie dla nich został wygenerowany. Co za tym idzie, możemy stwierdzać, badając skrót, czy dane, które posiadamy, są prawidłowe i nie zostały zmienione. Funkcje skrótu służą nam do tworzenia elektronicznych próbek DNA danych, na których działamy.

Patrząc z innej strony, funkcje skrótu są po prostu pewnymi przekształceniami matematycznymi, których dziedziną są dowolnej długości ciągi binarne, a wartościami ciągi binarne o określonej długości. Funkcje te dokonują swego rodzaju skrócenia danych dowolnej długości do danych określonej długości. Inaczej można powiedzieć, że funkcje te odwzorowują dane dowolnej długości na dane określonej długości.



Rys. 1. Funkcja skrótu

Powszechnie wykorzystywanym zastosowaniem funkcji skrótu są mechanizmy podpisu cyfrowego. Jak się domyślamy, podpis elektroniczny powinien oznaczać wszystkie dane, które zamierzamy podpisać. Inaczej mówiąc, nie powinna istnieć możliwość zmian danych podpisanych cyfrowo. Niedopuszczalna jest sytuacja, w której jedna ze stron zmienia na przykład kwotę umowy, a podpis pod dokumentem pozostaje niezmieniony. Rozwiązaniem tego problemu jest przetworzenie przez mechanizm podpisujący wszystkich danych, których podpis ma dotyczyć. Rozwiązanie niezwykle proste, ale jak pokazuje praktyka, niezwykle nieefektywne. Nieefektywność polega na tym, iż powszechnie używane schematy podpisów cyfrowych wykorzystują bardzo wolne prymitywy bazujące na kryptografii asymetrycznej. Nietrudno sobie wyobrazić, iż w tej sytuacji podpisywanie dużej ilości

danych może być niezwykle wolnym procesem, co w świecie dzisiejszego biznesu jest wręcz niedopuszczalne. Rozwiązaniem tego problemu jest zastosowanie funkcji skrótu w celu wygenerowania próbki DNA podpisywanych danych, a następnie podpisania nie samych danych, a tylko utworzonego skrótu. Sposób ten jest dużo bardziej efektywny i umożliwia podpisywanie dużych ilości danych w rozsądnym czasie.

Innym szeroko stosowanym wykorzystaniem funkcji skrótu jest uwierzytelnianie użytkowników i wiadomości, przechowywanie haseł, wyszukiwanie i przechowanie informacji. Możemy wyobrazić sobie sytuację, w której wartości skrótu są wykorzystywane w systemie do sprawdzenia, czy dany dokument został już zarejestrowany. Jeżeli funkcje skrótu nie zostałyby zastosowane, należałoby za każdym razem porównać nowy dokument ze wszystkimi przechowywanymi przez system. Funkcje skrótu są również powszechnie stosowane w celu weryfikacji integralności danych. Przykładem mogą być tutaj pliki ściągane z Internetu. Podczas transmisji mogą wystąpić zakłócenia lub plik może zostać celowo zmieniony przez osoby trzecie. Dlatego też w celu sprawdzenia, czy ściągnięty plik jest rzeczywiście tym, który chcieliśmy ściągnąć, oraz czy nie nastąpiły błędy transmisji, możemy wygenerować skrót i porównać go ze skrótem pliku poprawnego (skróty takie są coraz częściej zamieszczane w Internecie). Jeżeli skróty się zgadzają, to ściągnięty plik jest poprawny.

3. Definicja matematyczna

Funkcje mieszające powinny generować określonej długości skrót na podstawie dowolnej długości danych. Dowolną funkcję haszującą h można opisać za pomocą poniższego wzoru:

$$h: \ \Sigma^* \to \Sigma^n,$$

gdzie $\Sigma^* = \bigcup_{i \in N} \Sigma^i$, a $\Sigma^n$to n-elementowy wyraz zbudowany z liter alfabetu Σ.

Dla każdej funkcji h istnieje wiele par argumentów, dla których zwracane wartości są jednakowe. Jest to sytuacja nieunikniona z punktu widzenia teorii informacji. Wynika to z bardzo prostego faktu, którym jest stała długość wyjścia funkcji i nieograniczona długość wejścia. Posiadając parę wiadomości m_1 i m_2, kolizja zachodzi wtedy, gdy

$$h(m_1) = h(m_2).$$

Pomimo faktu istnienia wielu kolizji wiadomości dla dowolnych funkcji skrótu, podstawową miarą ich bezpieczeństwa jest odporność na kolizje rozumiana w tym sensie, że znalezienie pary wiadomości kolidujących powinno być praktycznie niewykonalne. Wymóg ten dotyczy wiadomości dowolnej długości, gdyż można zauważyć, że krótkie wiadomości składające się z kilku bitów również muszą być przetwarzane do n-bitowego skrótu. W praktyce krótkie wiadomości są dopełniane do pewnej ustalonej długości, a dopiero później przetwarzane.

Ogólnie funkcje skrótu mogą charakteryzować się następującymi własnościami:

- *Odporność na przeciwobraz* – oznacza, że dla dowolnej wartości funkcji y praktycznie niemożliwe jest znalezienie przeciwobrazu m takiego, że $y = h(m)$. Innymi słowy odporność na przeciwobraz implikuje jednokierunkowość.
- *Odporność na drugi przeciwobraz* – oznacza, że posiadając m, odnalezienie wiadomości m' takiej, że $h(m) = h(m')$, jest praktycznie niemożliwe. Własność ta bardzo często nazywana jest również słabą odpornością na kolizje.
- *Odporność na kolizje* – oznacza, że praktycznie niemożliwe jest znalezienie dwóch różnych wiadomości m_1 i m_2 takich, że $h(m_1) = h(m_2)$. Inną nazwą tej własności jest silna odporność na kolizje.

Wszystkie funkcje skrótu możemy ogólnie podzielić na dwie kategorie:

- *Jednokierunkowa funkcja mieszająca* (*one-way hash function*) – funkcja przetwarzająca wiadomości dowolnej długości i generująca skrót długości n. Obliczanie skrótu jest proste, a funkcja posiada własność odporności na przeciwobraz i odporności na drugi przeciwoobraz. Funkcje te posiadają przeznaczenie ogólne i nie jest zalecane ich stosowanie w celach kryptograficznych.
- *Odporna na kolizje funkcja mieszająca* (*collision-resistant hash function*) – funkcja przetwarzająca wiadomości dowolnej długości i generująca skrót długości n. Obliczanie skrótu jest proste, a funkcja posiada własność odporności na kolizje. Funkcje tego typu nazywana są również kryptograficznymi funkcjami skrótu lub silnie jednokierunkowymi funkcjami mieszającymi. Z uwagi na posiadane własności funkcje tego typu nadają się do zastosowań kryptograficznych.

Należy zauważyć, że odporność na kolizje daje nam możliwość używania funkcji bez specjalnej troski o to, czy przetwarzane wiadomości nie prowadzą do kolizji. Sama własność jednokierunkowości nie zapewnia nam tego komfortu.

4. Przekształcenia trójkątne

Funkcję $f(x)$ odwzorowującą, n-bitowe wejście na n-bitowe wyjście nazywamy przekształceniem trójkątnym lub w skrócie T-funkcją, jeżeli nie przenosi informacji od lewej do prawej strony, co oznacza, że bit numer i wyjścia ($[f(x)]_i$, gdzie LSB posiada numer 0, a MSB $n-1$) może zależeć jedynie od bitów wejścia $j = 0, \ldots, i$. Definicja ta może być rozszerzona w sposób naturalny na funkcje odwzorowujące kilka n-bitowych wejść na kilka n-bitowych wyjść. Klasę T-funkcji, w których i-ty bit wyjścia zależy jedynie od bitów wejścia $j = 0, dots, i-1$ nazywamy parametrami i oznaczamy greckimi literami alfabetu. Dzięki temu możemy każdą T-funkcję f traktować jako sparametryzowaną kolekcję odcięć bitowych odwzorowujących $[x]_i$ na $[f(x)]_i$, w których parametry opisują zależności wyjściowego bitu od poprzednich numerycznie bitów wejścia. Najprostszym przykładem takiej reprezentacji jest dodawanie dwóch wartości x, y: i-ty bit wyjścia jest XOR'em $[x]_i$ i $[y]_i$, a parametr opisuje przeniesienie z dodawania bitów numerycznie poprzednich.

Prostym przykładem T-funkcji są operacje bazowe wykonywane przez współczesne mikroprocesory operujące na słowach n-bitowych: dodawanie $(a+b(\mathrm{mod}\, 2\cdot n))$, odejmowanie $(a-b(\mathrm{mod}\, 2\cdot n))$, negacja $(\sim a(\mathrm{mod}\, 2\cdot n))$, mnożenie $a\cdot b(\mathrm{mod}\, 2\cdot n)$, dopełnienie bitowe, XOR $(a\oplus b)$, suma logiczna $(a\vee b)$ oraz iloczyn logiczny $(a\wedge b)$. Te osiem operacji maszynowych nazywać będziemy operacjami podstawowymi. Każda kompozycja operacji podstawowych jest również przekształceniem trójkątnym. W ogólności, wszystkie wielomiany modulo $2\cdot n$ są T-funkcjami. Należy zauważyć, że przesunięcie w lewo można by dodać do grupy operacji podstawowych, ponieważ operacja ta jest odpowiednikiem mnożenia przez 2, jednak nie można dodać do tego grona przesunięcia w prawo, gdyż nie jest ono T-funkcją.

Przekształcenia trójkątne umożliwiają łatwą weryfikację, czy dane odwzorowanie jest odwzorowaniem różnowartościowym, czy posiada własność pojedynczego cyklu oraz czy para odwzorowań tworzy tzw. kwadrat łaciński.

Różnowartościowe przekształcenia trójkątne o pojedynczym cyklu

Struktura cykli różnowartościowej T-funkcji jest niezwykle istotna w przypadku funkcji uaktualniających stan na przykład w szyfrach str umieniowych, ponieważ są one wykorzystywane w pojedynczym przetwarzaniu bardzo dużo razy i z tego powodu niepożądanym zjawiskiem jest „utknięcie" stanu w krótk im cyklu. Fakt łatwości testowania maksymalnej długości rejestrów LFSR zapewne stanowi jeden z głównych powodów

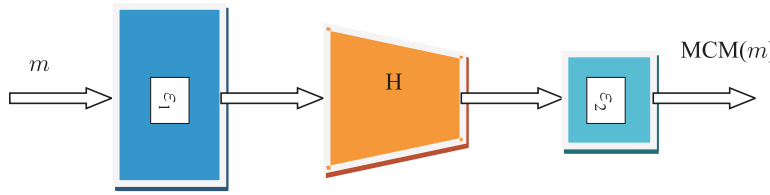
tak popularnego ich stosowania, niezależnie od kryptograficznych słabości wynikających choćby z liniowości funkcji zmiany stanu. Wykorzystując przekształcenia trójkątne, możemy tworzyć funkcje nieliniowe i niealgebraiczne posiadające własności pojedynczego cyklu i nieposiadające tak jak LFSR stanu zabronionego. Dzięki temu możemy zastąpić w naszych przyszłych konstrukcjach LFSR-y przekształceniami trójkątnymi, które wydają się zapewniać porównywalną efektywność przy znacznie lepszych własnościach kryptograficznych.

Jak już zostało wspomniane, dodatkową korzyścią jest fakt, że w LFSR-ach istnieje stan zabroniony (same zera), z którego rejestr nigdy nie wyjdzie, a przez to musimy być pewni, że po inicjalizacji algorytmu żaden LFSR nie jest właśnie w owym stanie, a w T-funkcjach stan taki nie występuje. Słabość ta została wykorzystana na przykład w atakach na A5/2 (algorytm szyfrowania w telefonach GSM). Posiadając odpowiednio wybraną T-funkcję, można zagwarantować, że wszystkie stany należą do cyklu o długości 2^n i nie istnieją stany zabronione, które powinny być unikane podczas inicjalizacji.

5. Metoda konstruowania funkcji skrótu MCM (MIX-Compress-Mix)

Od funkcji skrótu bardzo często oczekuje się zapewnienia bezpieczeństwa w różnego rodzaju aplikacjach, nawet gdy nie ma teoretycznych podstaw do wsparcia owych oczekiwań. Dla przykładu, SHA-1 jest używana jako odporna na kolizje funkcja skrótu oraz jako narzędzie inicjalizujące losowe wyrocznie (współczesne ataki pokazują, że oba zastosowania nie są odpowiednie). Lepsze bezpieczeństwo byłoby zapewnione przez wykorzystanie konstrukcji z udowodnianą własnościąć odporności na kolizje (np. poprzez wykorzystanie problemów trudnych obliczeniowo) i, jednocześnie, dającą gwarancję zachowania jak losowa wyrocznia. Funkcję skrótu spełniającą te wymagania nazywać będziemy zorientowaną na aplikacje. Niestety, jak pokazuje rzeczywistość, funkcje z udowadnialną odpornością na kolizje nie spełniają drugiego wymagania, gdyż struktura, na której bazują, pozwala na udowodnienie pierwszej własności, uniemożliwiając jednocześnie zachowanie się jak losowa wyrocznia. Konstrukcja MCM stanowi ogólny model, pozwalający na konstruowanie funkcji skrótu zorientowanych na aplikacje. Konstrukcja ta w połączeniu z funkcją skrótu odporną na kolizje pozwala tworzyć funkcje spełniające obie wymagane przez nas własności w modelu rzeczywistym i nieodróżnialne od losowej wyroczni w modelu idealnym.

Ogólna konstrukcja MCM (Mix-Compress-Mix) została stworzona jako alternatywa do obecnie znanych metod, umożliwiająca tworzenie funkcji skrótu zorientowanych na aplikacje.



Rys. 2. Konstrukcja MCM

Idea konstrukcji MCM jest bardzo prosta. Pierwszym krokiem jest mieszanie ε_1, następnie kompresja H z wykorzystaniem funkcji odpornej na kolizje i na końcu ponownie mieszanie ε_2. H i ε_1 operują na danych wejściowych dowolnej długości. Przekształcenia ε_1 i ε_2 muszą być deterministyczne i publicznie dostępne. Dodatkowo wymagając, aby ε_1 i ε_2 były iniekcjami, kolizje przeciwko MCM oznaczają kolizje przeciwko H. Żadne dodatkowe założenia co do ε_1 i ε_2 nie są wymagane w celu zapewnienia odporności na kolizje.

Konstrukcje MCM spełniają drugie kryterium, czyli zachowują się jak losowe wyrocznie, gdy ε_1 i ε_2 są pseudolosowymi iniekcyjnymi wyroczniami (PRIO) i funkcja H jest prawie regularna (zbiór przeciwobrazów dla dowolnego wyjścia nie jest zbyt liczny). Innymi słowy, ε_1 i ε_2 powinny zachowywać się jak losowe wyrocznie zachowujące własność iniekcji. Poprzez „zachowanie jak" rozumiemy wykazanie nieodróżnialności zgodnie z modelem przedstawionym przez Maurer'a.

Intuicyjne wyjaśnienie, dlaczego MCM pozwala tworzyć funkcje zorientowane na aplikacje jest proste: kroki miksujące niszczą wszystkie relacje wejścia-wyjścia kroku kompresującego.

Konstrukcja MCM pozwala na podział pracy na etapy, w których konstruowane będą prymitywy niezależnie od siebie. Na początku projektujemy funkcję gwarantującą silną odporność na kolizje. Następnie tworzymy przekształcenia injekcyjne, które będą niszczyć strukturę danych wejściowych. Konstrukcja MCM i towarzysząca jej teoria łączenia pozwala połączyć stworzone prymitywy celem otrzymania funkcji zorientowanej na aplikacje.

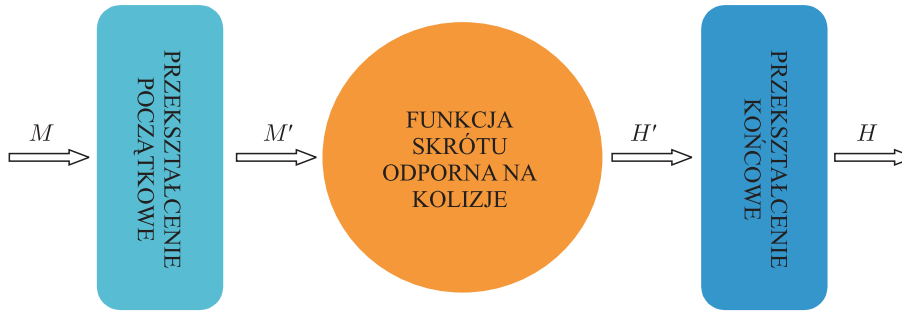
6. Projekt funkcji skrótu bazującej na konstrukcji MCM i korzystającej z przekształceń trójkątnych

Szkielet funkcji skrótu oparty został o konstrukcję Thomasa Ristenparta i Thomasa Shrimptona MCM. Dzięki temu, rozpatrując moją konstrukcję w modelu idealnym, otrzymałem funkcję nieodróżnialną od losowej wyroczni.

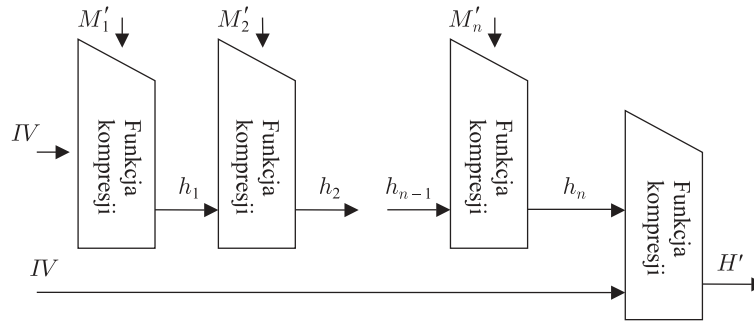
Budując odporną na kolizję funkcję skrótu, bazowałem na pracy *A new Design Criteria for Hash-Functions*. Składową funkcję utworzyłem zgodnie z przedstawioną w pracy metodą poprawy własności konstrukcji MD o nazwie HMAC.

Tworząc funkcję kompresji, wykorzystałem metodykę przedstawioną w pracach Klimova i Shamira na temat przekształceń trójkątnych. Zaproponowana przeze mnie funkcja bazuje na ortogonalnych kwadratach łacińskich.

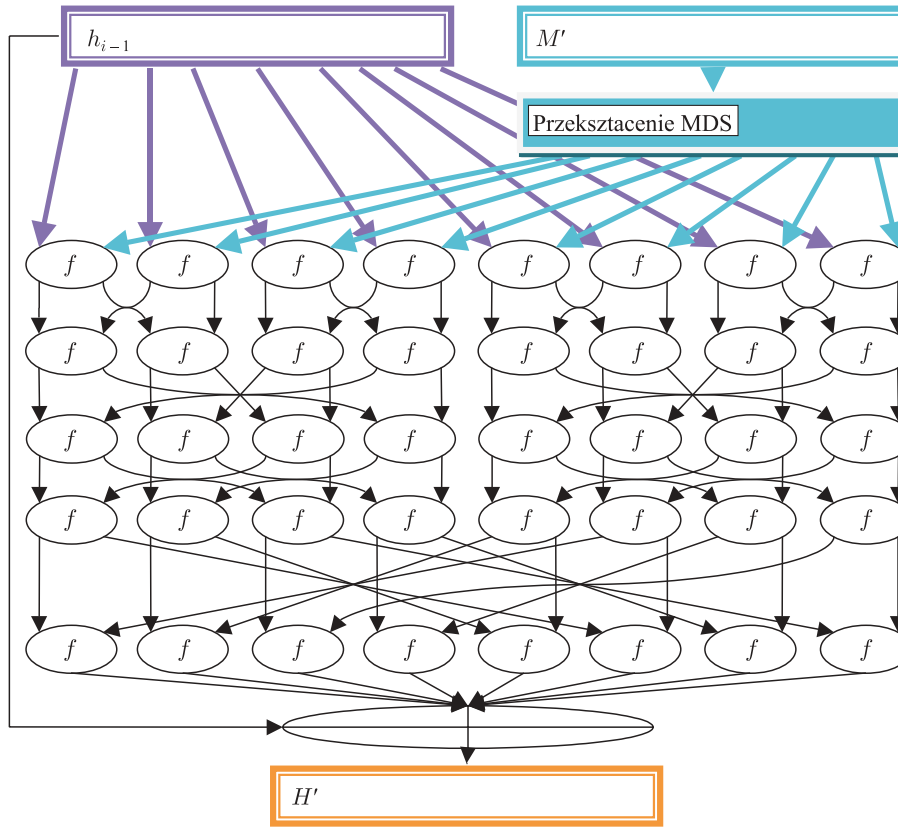
7. Konstrukcja



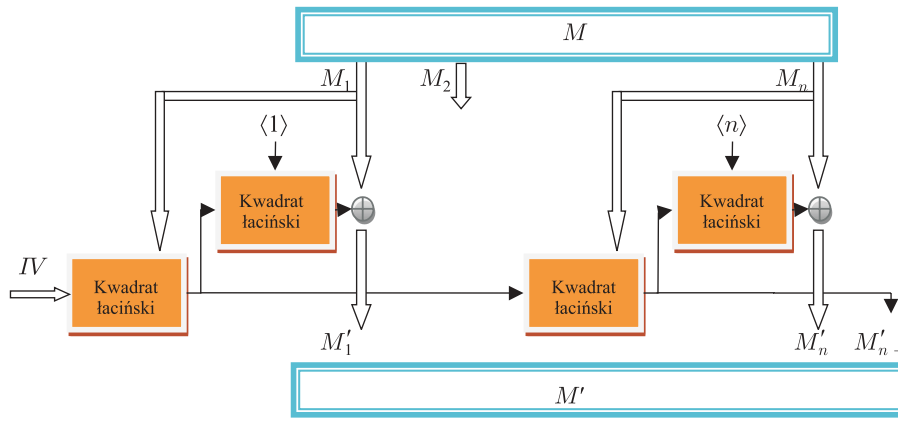
Rys. 3. Szkielet funkcji skrótu



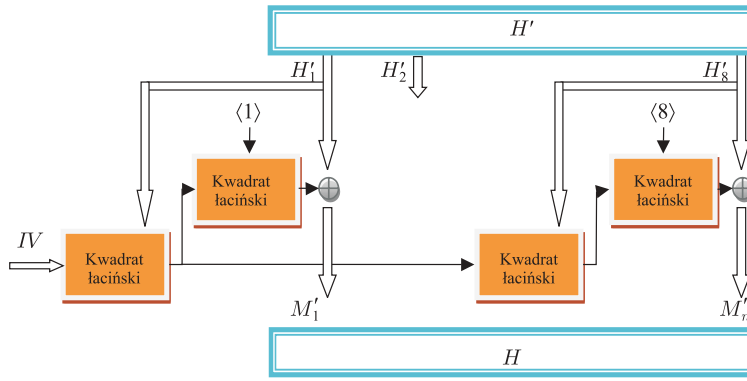
Rys. 4. Szkielet składowej funkcji skrótu



Rys. 5. Pojedyncza iteracja składowej funkcji skrótu



Rys. 6. Przekształcenie początkowe



Rys. 7. Przekształcenie końcowe

8. Podstawowe własności konstrukcji

Bazując na szkielecie MCM, zapewniłem, przy odpowiednim doborze przekształceń mieszających, że analiza związków bitów wejściowych i wyjściowych będzie niezwykle skomplikowana. Dodatkowo konstrukcja MCM zapewnia własność nieodróżnialności wyjścia funkcji od wyjścia z losowej wyroczni (w modelu idealnym).

Składową funkcję skrótu odporną na kolizje oparłem na najpowszechniej dzisiaj stosowanej konstrukcji – schemacie Merkle-Damgard. Dzięki temu wystarczyło, w celu osiągnięcia własności odporności na kolizje, żeby wykorzystana funkcja kompresji była odporna na kolizje. Dodatkowo zastosowałem schemat iterowania znany z konstrukcji typu HMAC. Dzięki temu uodporniłem składową funkcję skrótu na znane ataki ogólne (w tym na atak rozszerzania wiadomości).

Funkcja kompresji, która nadawałaby się do wykorzystania w tej konstrukcji, musi być odporna na kolizje. Jest to wymagane, ponieważ dzięki temu również składowa funkcja skrótu będzie odporna na kolizje. Dlatego właśnie wykorzystałem pomysł wprowadzony wraz z funkcją skrótu bazującą na FFT – multipermutacje. Zastosowana w projekcie funkcja kompresji bazuje na sieci multipermutacji. Dodatkowo każdy przetwarzany przez nią blok wiadomości jest poddawany przekształceniu typu MDS (Maximum Distance Separable). Dzięki temu otrzymujemy idealną dyfuzję bitów wejściowych. Na zakończenie przetwarzania wyjście z sieci multipermutacji jest poddawane operacji XOR z wejściem skrótu do funkcji kompresji. Operacja ta bardzo utrudnia analizę funkcji kompresji podczas szukania przeciwobrazów.

Poprzez wykorzystanie opisanych powyżej metod konstruowania i odpowiedni dobór konkretnych przekształceń, otrzymujemy funkcję skrótu odporną na kolizje i nieodróżnialną od losowej wyroczni.

Wszystkie funkcje składowe wykorzystane w projekcie powstały i zostały przeanalizowane zgodnie z metodyką przedstawioną wraz z pracami na temat przekształceń trójkątnych.

9. Podsumowanie

Przedstawiona przeze mnie funkcja skrótu wykorzystuje opisane wcześniej metody projektowania. Dzięki temu posiada własności, które posiadają tamte schematy. Dlatego też mój projekt charakteryzują dwie bardzo ważne cechy – odporność na kolizje oraz zachowanie się jak losowa wyrocznia. Zgodnie z tym, o czym pisałem wcześniej, funkcję skrótu spełniającą obie te własności nazywamy funkcją zorientowaną na aplikacje. Innymi słowy rozwiązanie to nadaje się praktycznie do każdego rodzaju zastosowania bez narażania bezpieczeństwa systemu.

Moja propozycja wymaga jeszcze wielu różnego rodzaju badań i analiz. Wynika to z faktu zastosowania przekształceń, które nie zostały poddane wnikliwej analizie. Przedstawiona przeze mnie funkcja spełnia zakładane własności pod warunkiem, że wykorzystane w niej przekształcenia składowe charakteryzują się odpowiednimi cechami. Można zatem zauważyć, że zaproponowaną przeze mnie funkcję można potraktować jako kolejną metodę projektowania, w której zadaniem jest znalezienie i zastosowanie odpowiedniej jakości przekształceń oraz dobór odpowiedniej konfiguracji wielkości danych, na jakich pracujemy, celem uzyskania możliwie efektywnego rozwiązania zgodnego z oczekiwaniami.

Praca naukowa finansowana ze środków na naukę w latach 2009–2011 jako projekt rozwojowy Nr O R00 0043 07

Artykuł wpłynął do redakcji w dniu 30.05.2011 r. Zweryfikowaną wersję po recenzji otrzymano w sierpniu 2011 r.

LITERATURA

[1] J-S. Coron, Y. Dodis, C. Malinaud, P. Puniya, *A new Design Criteria for Hash-Functions*, NIST's First Cryptographic Hash Workshop, 2005.

[2] A. Klimov, A. Shamir, *Cryptographic Applications of T-functions*, Selected Areas in Cryptography, SAC 2003, LNCS 3006: 248–261, Springer-Verlag, 2003.

[3] A. Klimov, A. Shamir, *Applications of T-functions in Cryptography*, PhD thesis, Weizmann. Institute of Science, 2004.

[4] V. Lyubashevsky, D. Micciancio, Ch. Peikert, A. Rosen, *Provably Secure FFT Hashing*, NIST Second Cryptographic Hash Function Workshop, 2006.

[5] J. Pieprzyk, T. Hardjono, J. Seberry, *Teoria bezpieczeństwa systemów komputerowych*, Helion, 2003.

[6] J. Pieprzyk, B. Sadeghiyan, *Design of Hashing Algorithms*, Springer-Verlag, 1993.

[7] T. Ristenpart, T. Shrimpton, *Building Application-Agile Hash Functions: the MCM Construction*, ECRYPT Workshop on Hash Functions, 2007.

[8] C.P. Schnorr, S. Vaudenay, *Parallel FFT-Hashing*, Fast Software Encryption, 149–156, Cambridge Security Workshop, 2003.

[9] http://csrc.nist.gov/groups/ST/toolkit/secure_hashing.html

[10] http://www.nist.gov

[11] http://www.wikipedia.com

M. GLET

MCM and *T*-functions – new way of constructing hash functions

Abstract. This paper presents a new familly of cryptographic hash functions based on Mix-Compess-Mix schema, multipermutation and T-functions. This new familly has very good cryptographic properties and can be impelmented in a very effective way in multithreaded environments.

Keywords: hash function, multipermutation, Mix-Compress-Mix, T-functions, MD